HW-5

Q1a)

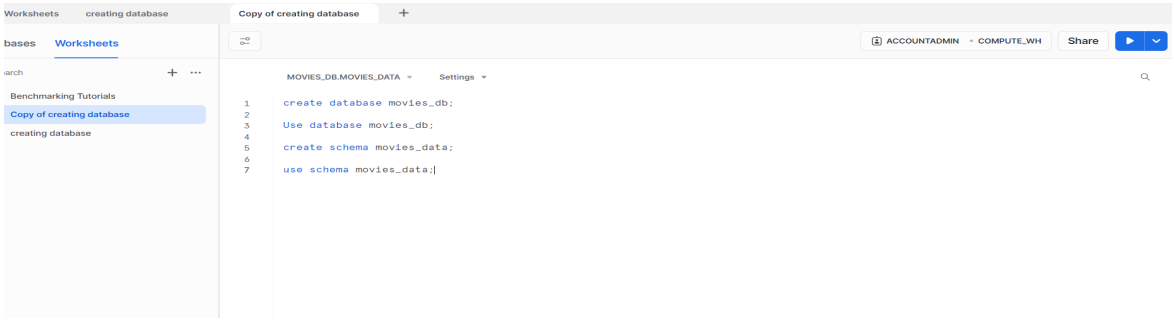
I am using Snowflake in the first part. I did not use it in my project that is why I explored Snowflake for this question.

The dataset I used is a movie dataset. Below is the drive link for the database.

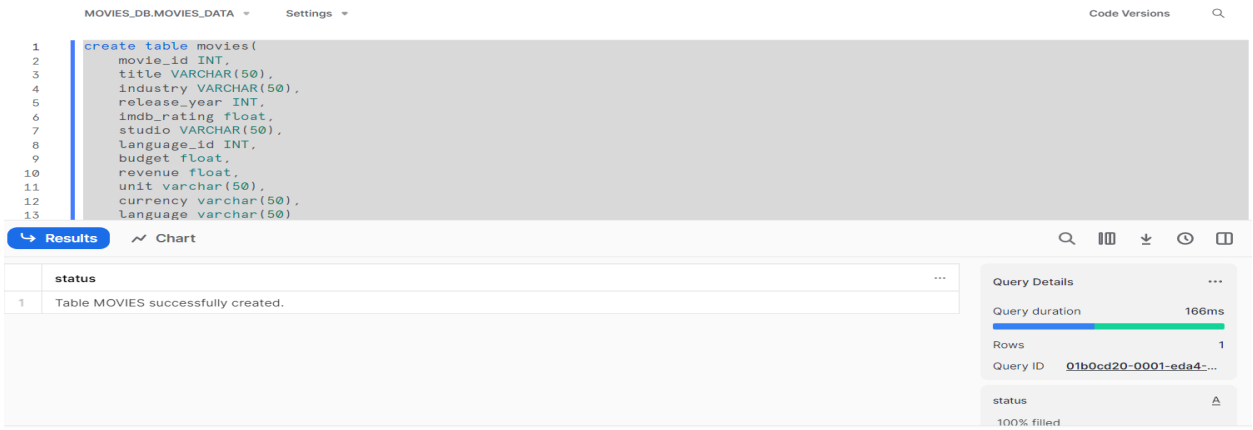
https://docs.google.com/spreadsheets/d/1JuNNgR50-815QoptzEIm4RYZrJE9eQv0/edit?usp=sharing&ouid=105974981587456835079&rtpof=true&sd=true

- Database creation and schema creation

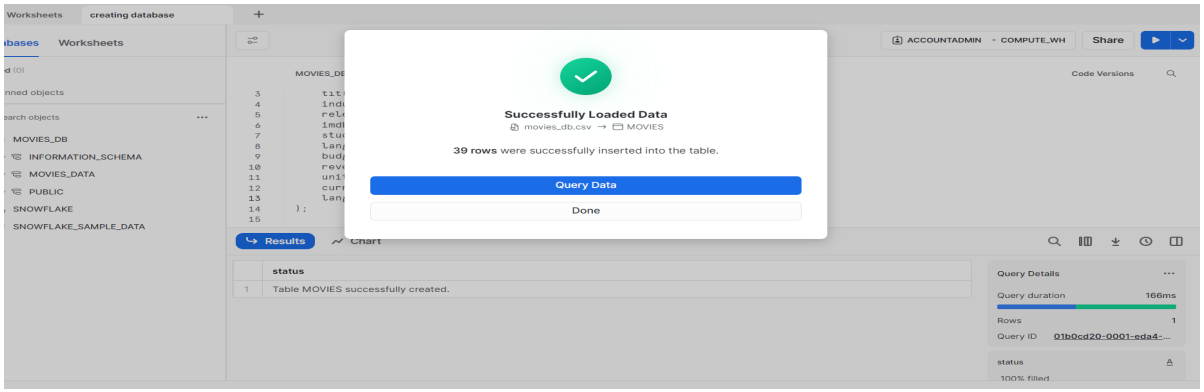
  The very first step I performed is database creation. I created a dataset named movies_db.



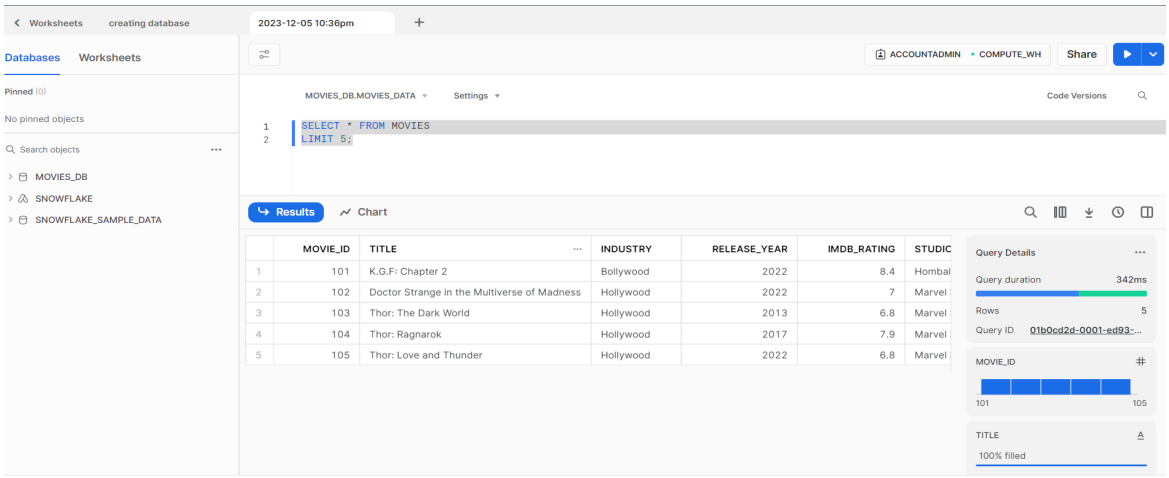
- Created a table to load the dataset which I downloaded



- Loaded the dataset into the movies table which I created and the snippet is above about the creation of movies table.

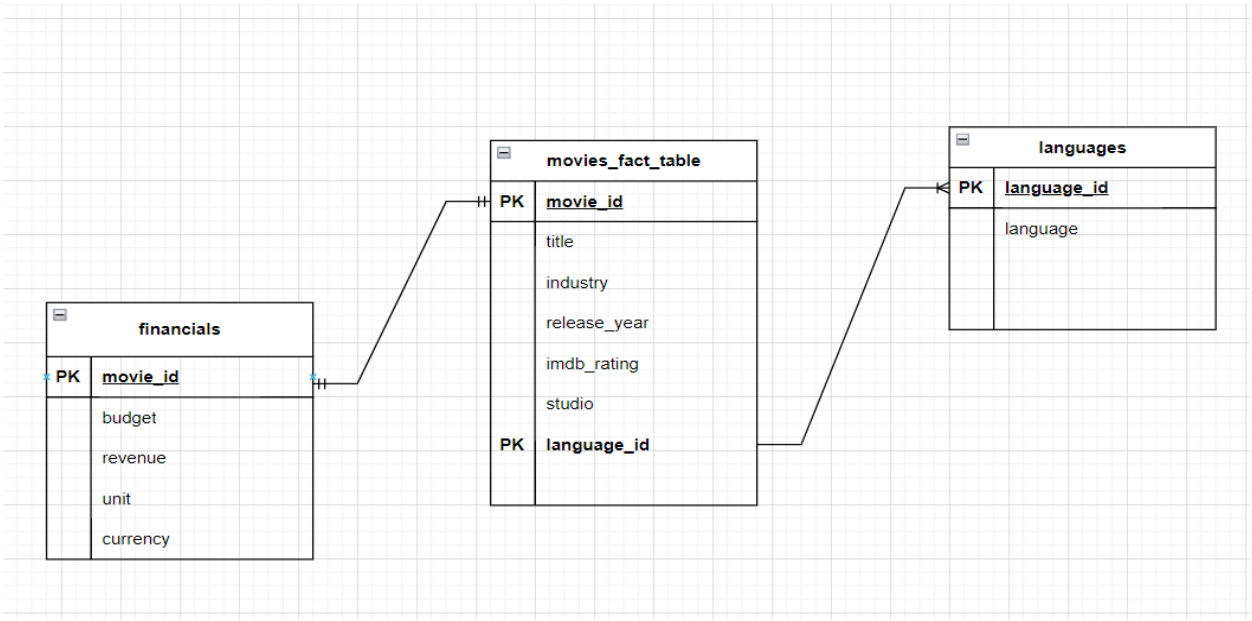


- Checking if the data is properly loaded or not.

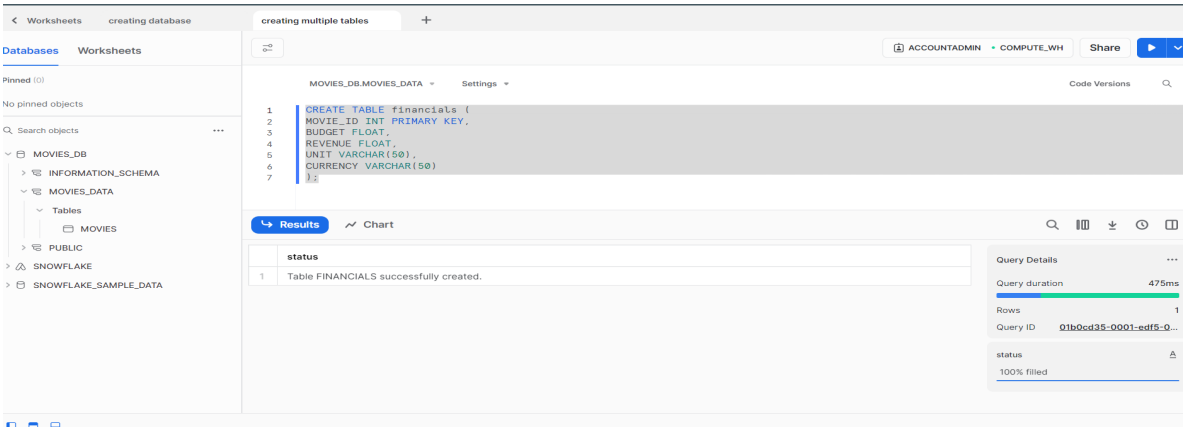


- Below is the star schema of the dataset. It is only possible to create two dimension tables and not more than that. If I create different dimension tables for currency or unit it will only have redundant data. Instead I kept all that information in the financials dimension table.

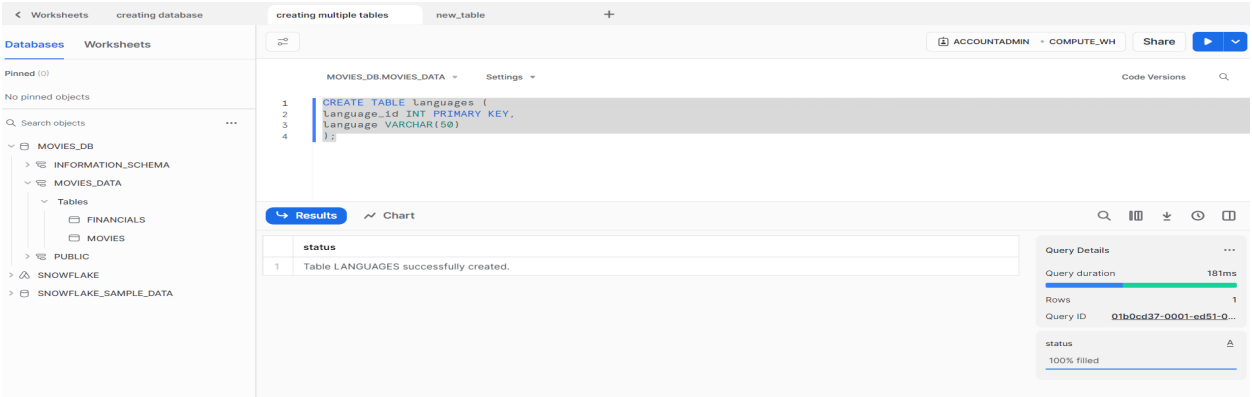
- Also In languages dimension table I tried to save a lot of space as I mapped a single langiage_id to each language and the maximum number of rows required here are only 8, whereas in the main table this data was filled in every row.



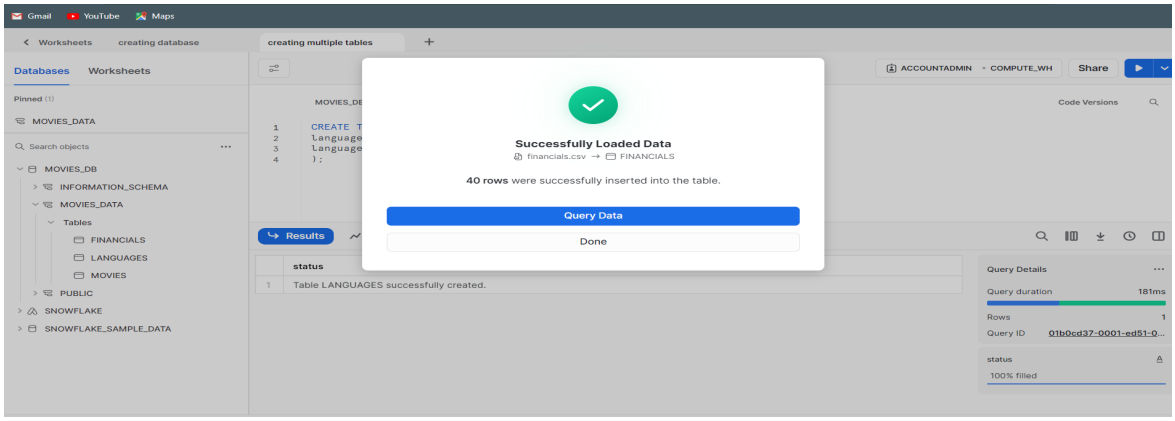
- Created dimension table financials according to the star schema which I have created.

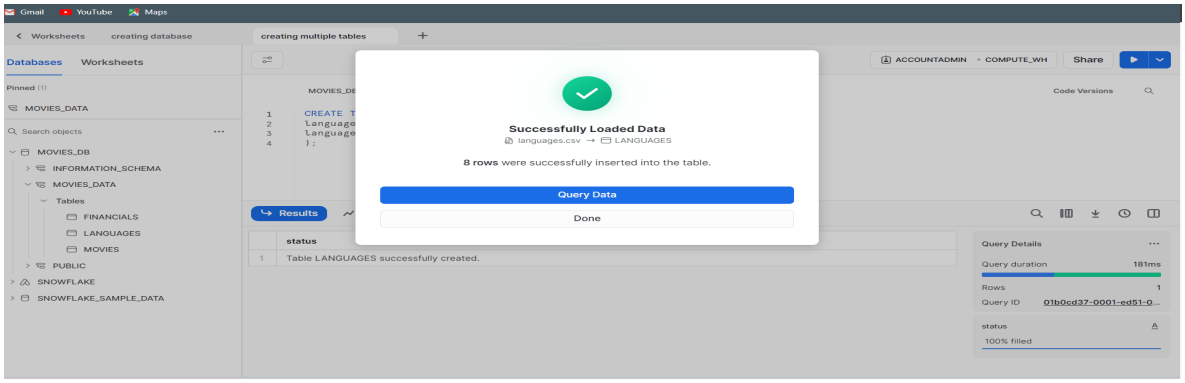


- Created another dimension table of languages according to the star schema which I have created.

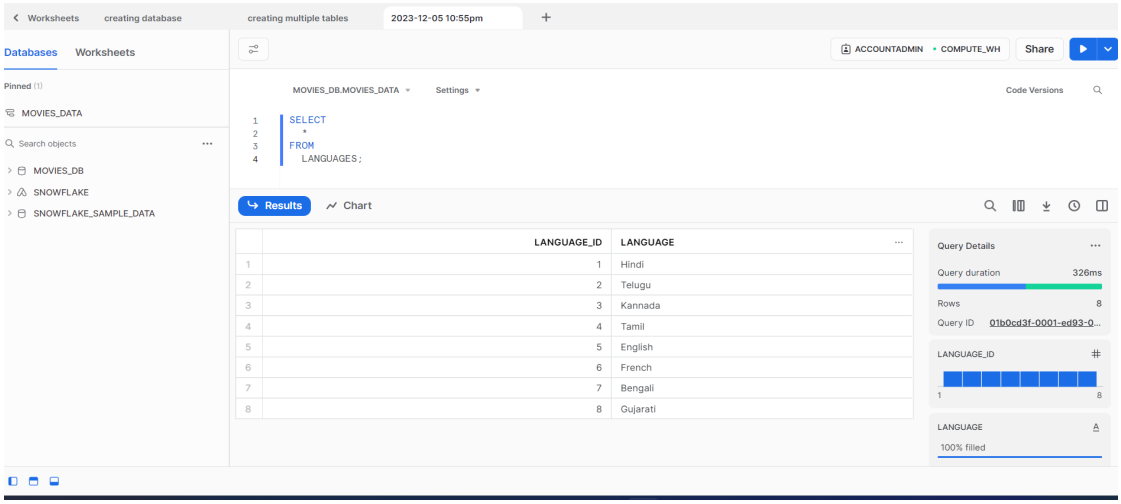


- These two below are the screenshots of loading data into newly created dimension tables tables.

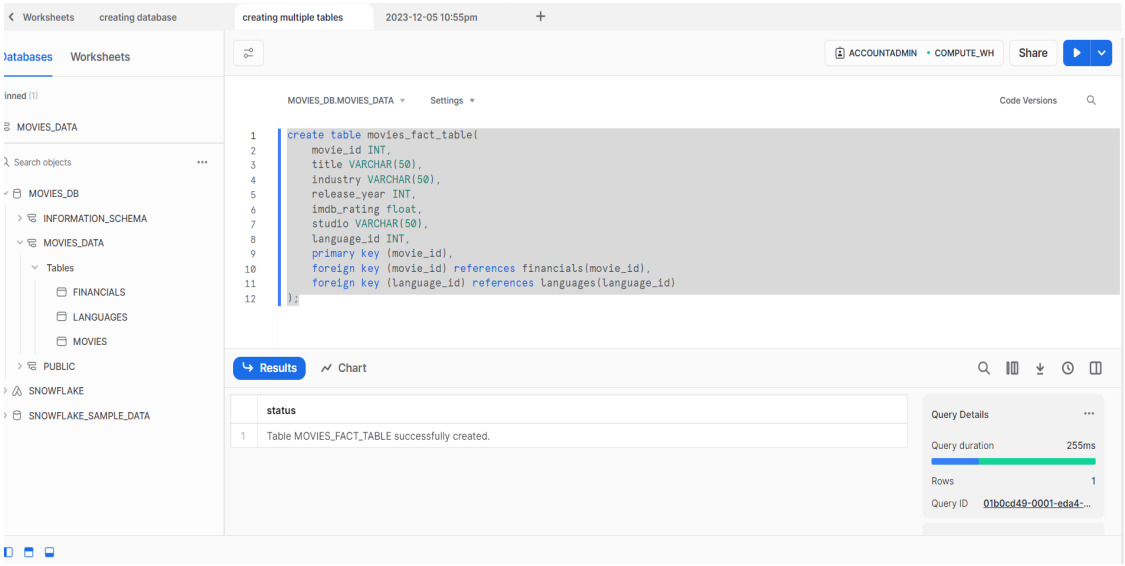




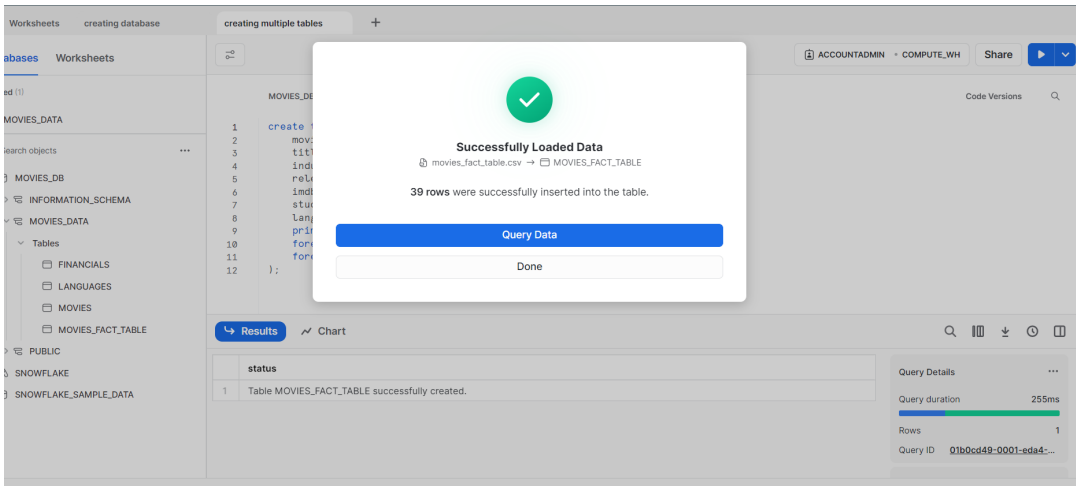
- Below screenshot is just of verification that whether the data has properly been loaded or not.

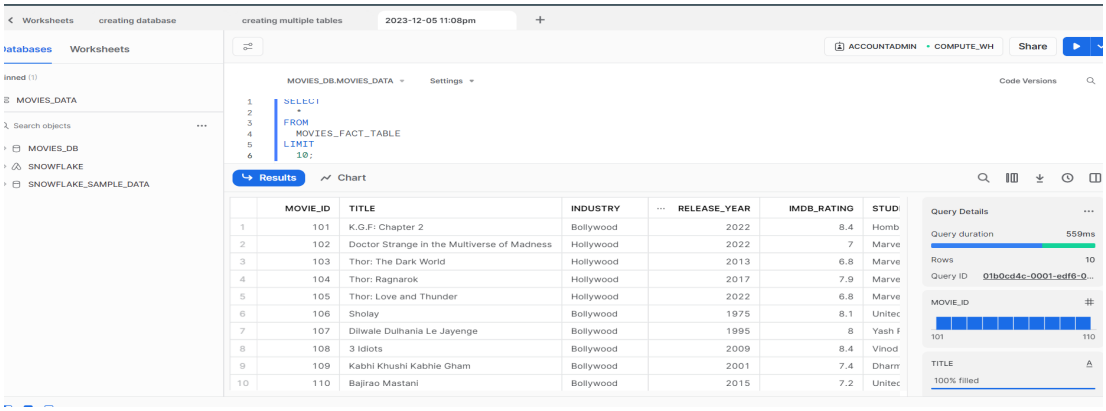


- Now I have created the fact table where movie_id is the primary key and it has 2 dimension tables named financials and languages.



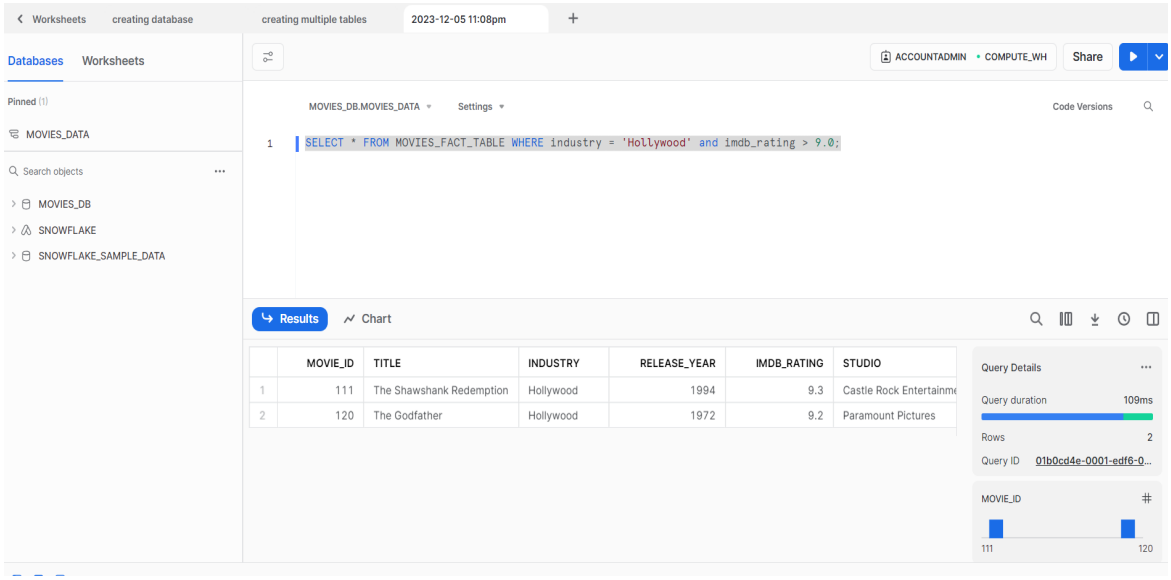
- Below are the screenshots which verifies that data has been successfully loaded into the fact table.



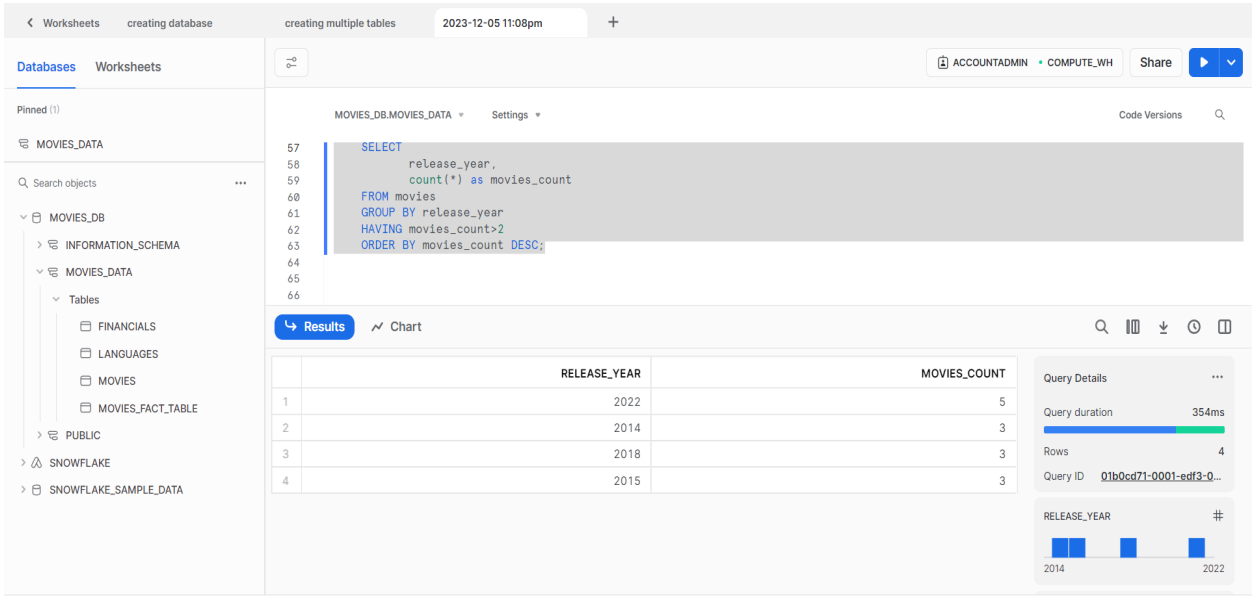


Analytics part(Queries)

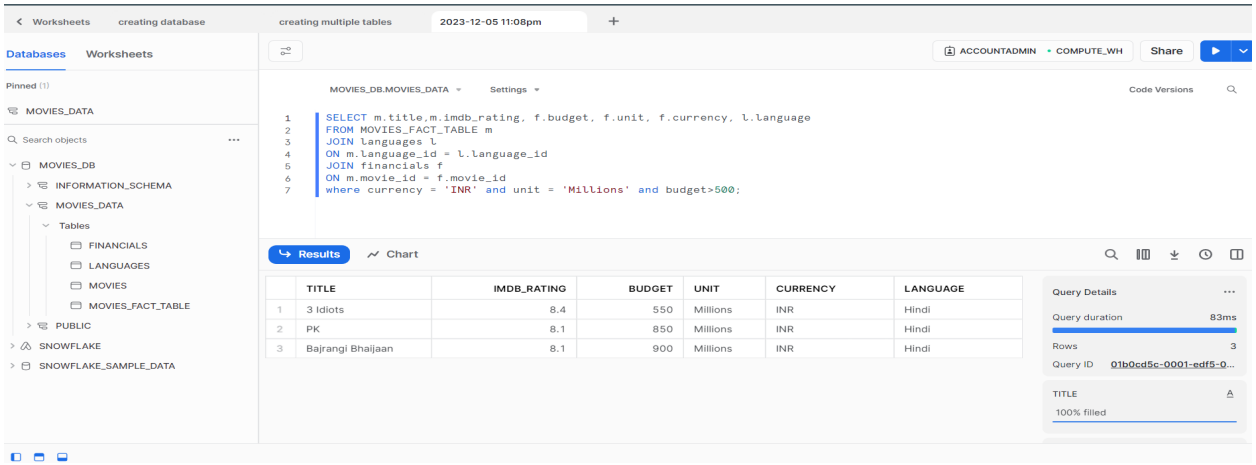
1)Below is the basic query which I tried to find out about the movies released by hollywood and their imdb rating is more than 9.



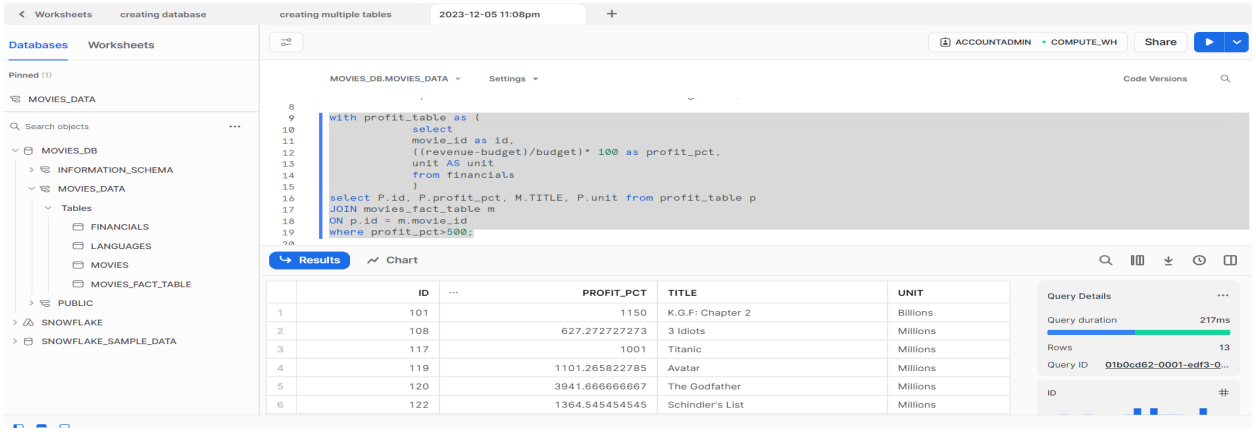
2) Below I tried to print all the years where more than 2 movies were released



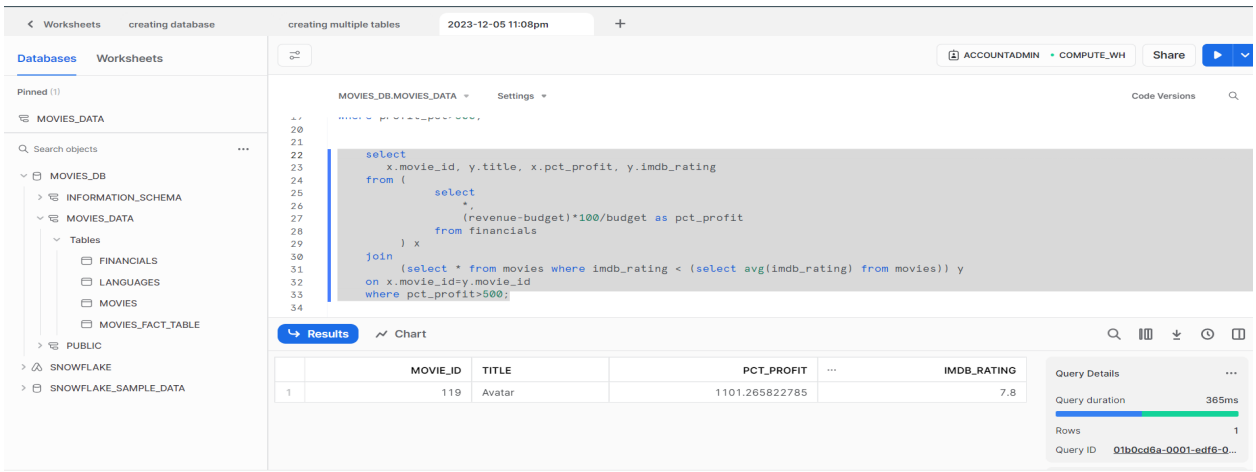
3)Below I joined all the three tables and printed the movies whose currency is ‘INR’ and Unit is ‘Millions’ and the budget is more than 500 where the budget is calculated on the basis of unit unit.



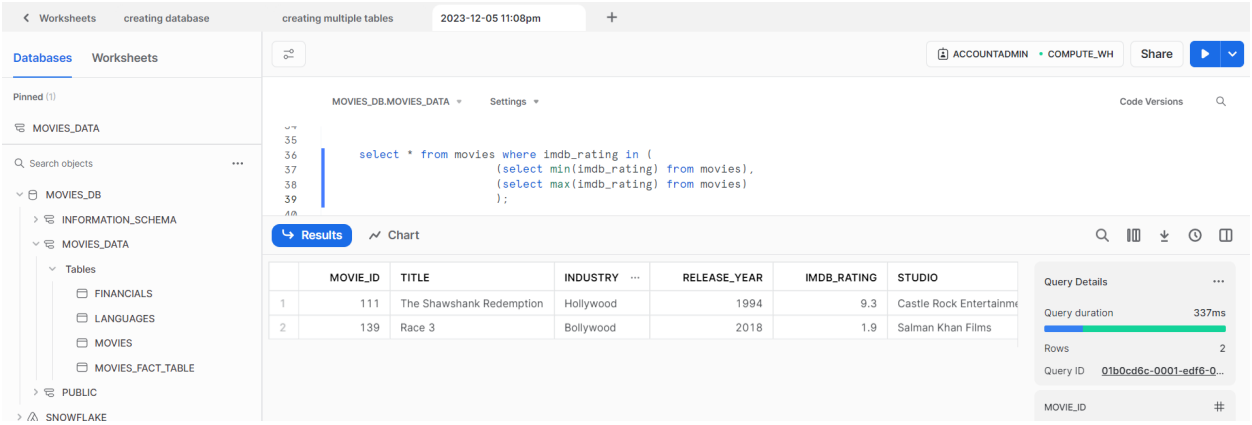
4)Below I have used Common Table Expression(CTE’s) and printed movies that produced 500% profit.



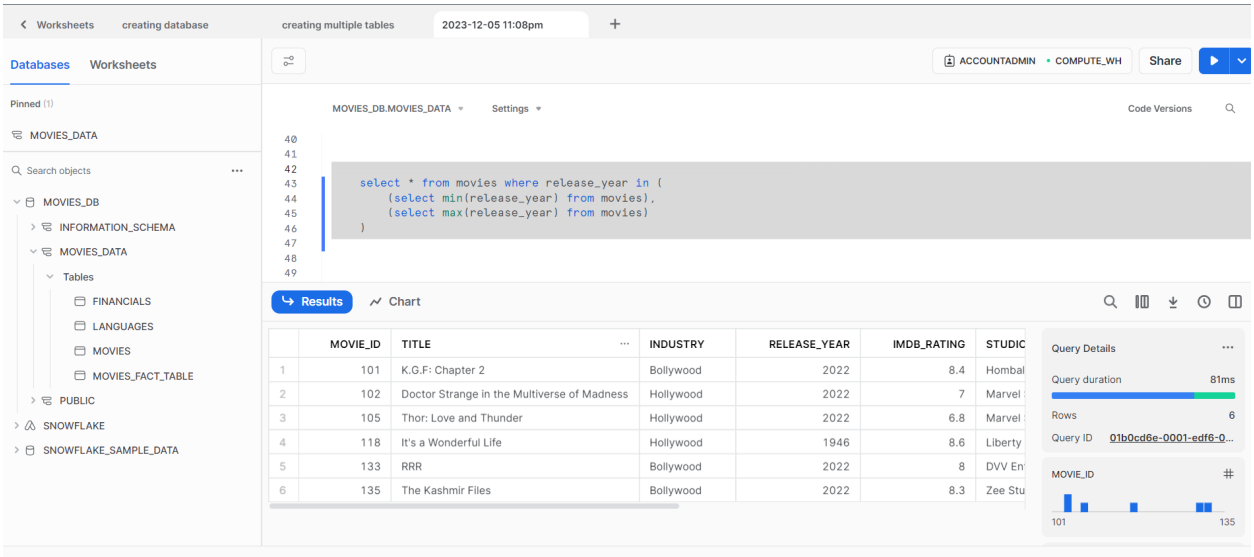
5)Below I have printed movies that produced 500% profit and their rating was less than average rating for all movies.



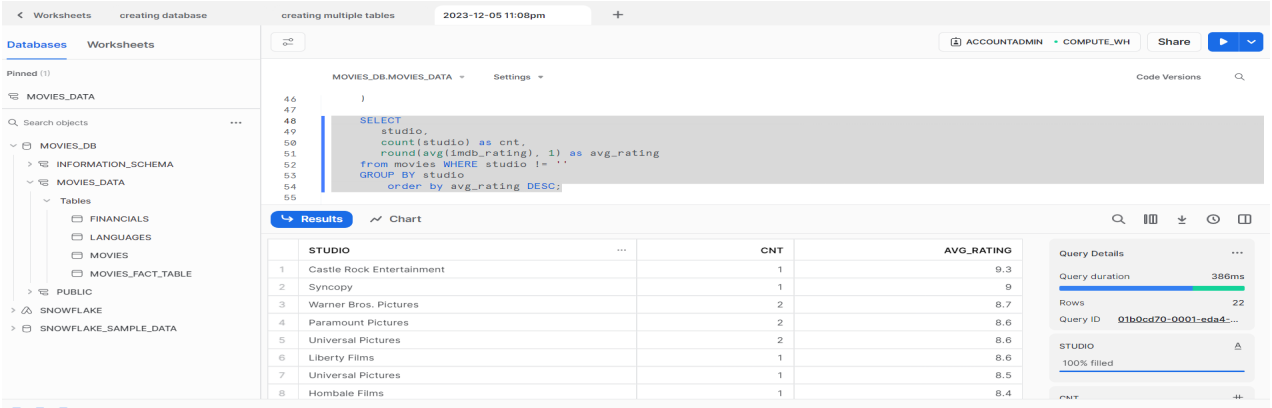
6)Below I have printed Movies with highest and lowest ratings from the dataset.



7) Below I have printed all the movies with minimum and maximum release_year.



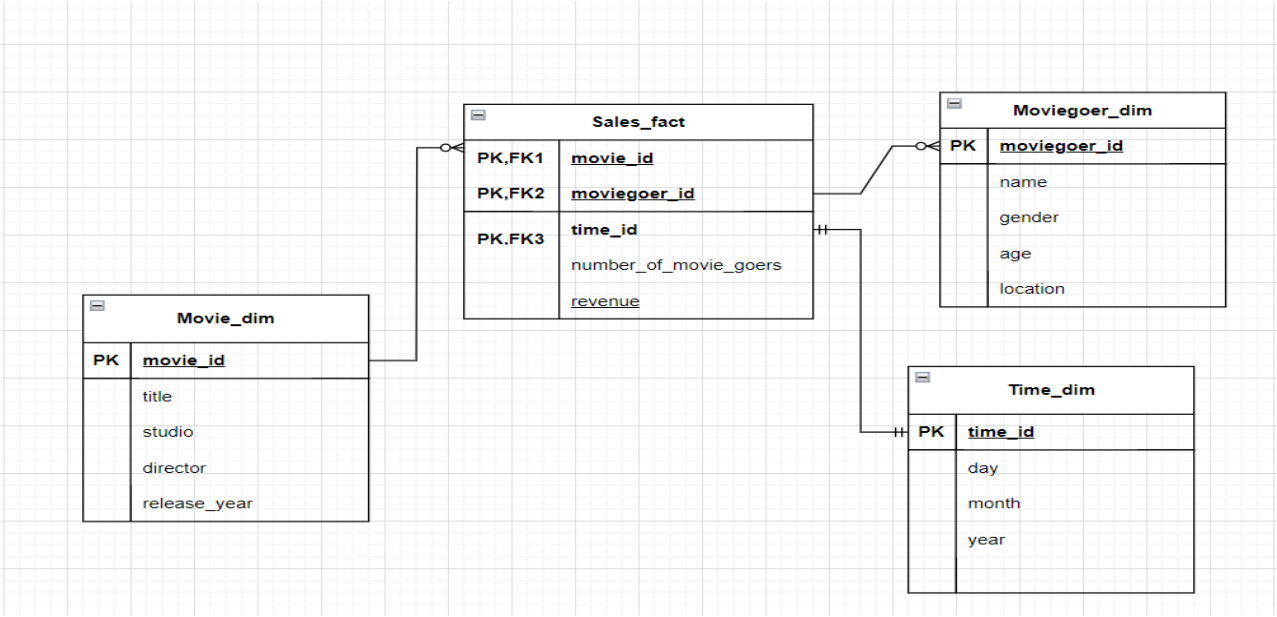
8) Below I have printed the average rating of movies per studio and also ordered them by average rating in descending format.



Q1)b) Below is the link to the blog written on Medium:

https://medium.com/@yeolevishal11/insights-from-a-movies-dataset-using-snowflake-d50b809b499d

Q2)
a)



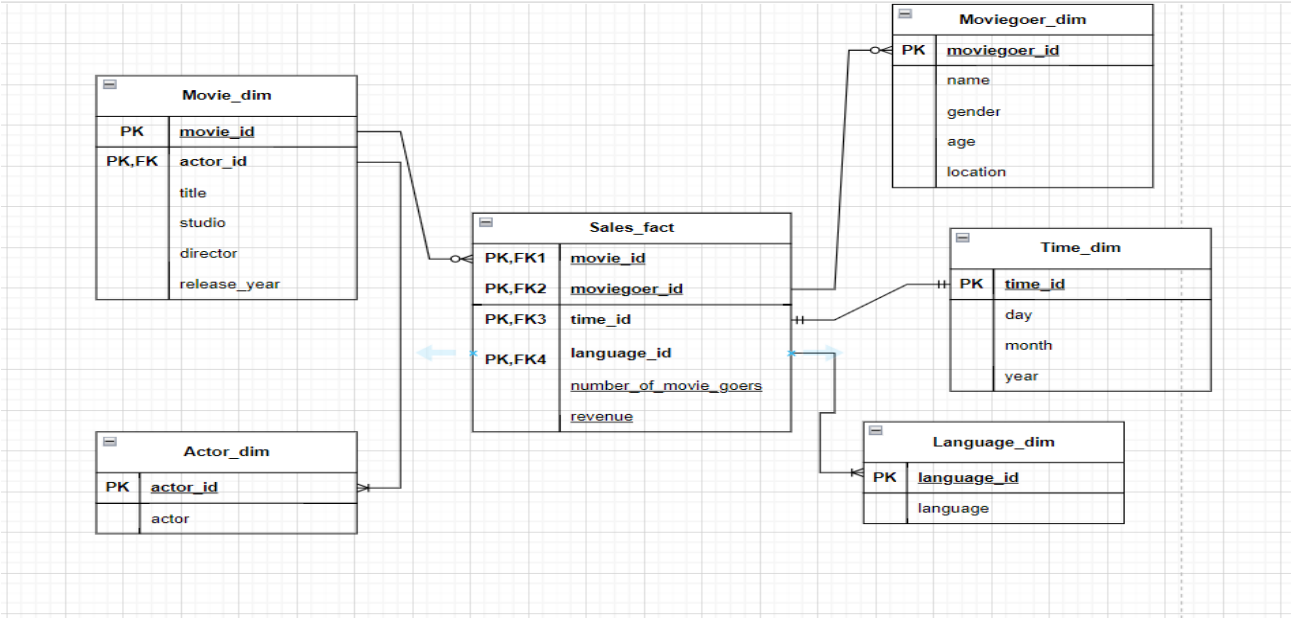
b)Total revenue collected through ticket sales by each movie in the pandemic year, 2020.

```
SELECT movie_dim.title, SUM(Sales_fact.revenue) AS total_revenue
FROM Sales_fact s
JOIN movie_dim m
ON s.movie_id = m.movie_id
JOIN Time_dim t
ON s.time_id = t.time_id
WHERE t.year = 2020
GROUP BY m.title;
```

c)

```
SELECT movie, SUM(ticket_price) AS total_revenue
FROM ticket_sales
WHERE year = 2020
GROUP BY movie;
```

d) This is the extended schema that is Snowflake schema where I have added language table and the actor table. Accordingly I performed a couple of join queries considering the newly added tables which are listed below.



- Query to find revenue for each language of a movie
  SELECT M.title AS Movie_Name , L.language AS language , SUM(S.Revenue) AS Revenue
  FROM Movie_dim M
  LEFT JOIN Sales_fact S USING(movie_id)
  LEFT JOIN Language_dim L USING(language_id)
  GROUP BY 1,2
  ORDER BY 3 DESC;

- Query to find the most earning move for each actor
  SELECT A.actor AS Actor, K.title AS Movie_Name
  FROM (
  Select P.actor_id , P.title , RANK() OVER(PARTITION BY P.actor_id,P.title ORDER BY P.Revenue DESC) Rank
  FROM (
  SELECT M.actor_id AS actor_id , M.title AS title , SUM(S.Revenue) AS Revenue
  FROM Movie_dim M
  LEFT JOIN Sales_fact S USING(movie_id)
  GROUP BY 1,2) P) K
  WHERE K.Rank = 1;

Q3)

a)

Columnar databases find application in data warehouses where enterprises funnel extensive data from various origins for Business Intelligence analysis. The efficiency of query performance in column-oriented databases is heightened due to the design that maintains data proximity, minimizing seek time.

Utilizing a columnar database addresses the constraints inherent in conventional relational databases, positioning columnar databases as the forthcoming frontier in business intelligence. In a database housing millions of records, a columnar database grants access to the most pertinent elements, enhancing query speed significantly. While traditional relational databases continue to serve as comprehensive data sources, the architecture of columnar databases simplifies the analysis of the entire dataset. The organizational structure of data in columnar databases leads to swifter outcomes and more effective analysis. Columnar databases excel at queries which involve few columns or aggregation queries against vast amounts of data.

The key observations I found from my study are:

- Numerous columnar databases employ parallel processing to horizontally expand, allowing them to manage expanding datasets and meet growing analytical requirements. This scalability proves crucial for businesses constantly accumulating data over time.
- Columnar databases can seamlessly adapt schemas by introducing new columns without disrupting existing structures or queries. This flexibility proves valuable in data warehousing, where schemas often undergo changes over time. An example of this adaptable columnar design is evident in ClickHouse, empowering organizations like Cloudflare to manage shifting data requirements.
- Materialized views empower columnar databases to pre-compute and store results for common analytical queries, significantly enhancing performance. Exasol incorporates this feature, utilized by some companies to expedite analytics.
- Columnar databases seamlessly integrate with popular business intelligence, visualization, and analytics tools, simplifying the extraction of insights from data.
- The use of columnar storage enhances the speed of query execution, especially for analytical queries, by minimizing the data processed. This efficiency stems from the storage of data in columns rather than rows, aligning seamlessly with the structure of analytical queries.
- Specialized compression techniques tailored to the storage format are frequently employed by columnar databases. This results in decreased storage needs and faster data retrieval, both vital for applications in data warehousing that manage extensive data volumes.

b)

- IBM Cognos
  IBM Cognos is an analytics tool with full potential of the data with AI powered automation. The AI-powered assistant is consistently accessible – articulate our data requirements, and Cognos Analytics will craft compelling data visualizations for us. If we express a query or hypothesis for testing, AI will do the rest of the part and it will draw all the possible necessary insights.
  This is a transition to the next era of Business Intelligence, featuring AI capabilities that not only deliver a precise, reliable, and comprehensive view of the business but also forecast future developments, predict outcomes, and elucidate the reasons behind them. From the IBM Cognos website I can see that they have recently launched Cognos Analytics 12.0 which is the more powerful version.
  I also think that with the development of such powerful AI analytical tools the jobs in data engineer and data scientist fields will increase whereas data analyst jobs will substantially decrease. It is because most of the analytical and visualization part is performed by AI powered tools like IBM Cognos.

- Microsoft SQL Server Analysis Services (SSAS):
  Analysis Services is used for corporate analytics and decision assistance as an analytical data engine (VertiPaq). It provides complex semantic data models for client applications and business reports, including Excel, Power BI, Reporting Services reports, and a range of data visualization tools.
  SQL Server Analysis Services supports multidimensional models, data mining, tabular models across compatibility levels (depending on version), and Power Pivot for SharePoint. It may be installed as an on-premises server or virtual machine instance. Typically, the standard workflow involves the installation of a SQL Server Analysis Services instance, the development of a tabular or multidimensional data model, deployment of the model as a database to a server instance, processing the database to populate it with data, and subsequently granting permissions to facilitate data access. Once prepared, any client application supporting Analysis Services as a data source can access the data model.

- Oracle OLAP
  A top-notch multidimensional analytic engine called Oracle OLAP is built into Oracle Database 12c. Oracle OLAP cubes produce results faster than the speed of thought by performing complex computations with straightforward SQL queries. When OLAP cubes are deployed as materialized views, the exceptional query performance can be transparently utilized to improve summary query performance against detail relational tables. Oracle OLAP enables centralized data and business rule management on a safe, scalable, and enterprise-ready platform because it is integrated into Oracle Database 12c. Analytical measures such as time-series calculations, financial models, forecasts, allocations, regressions, and more can be easily produced with Oracle OLAP. Almost

any analytical calculation need can be satisfied by simply combining hundreds of analytic functions into custom functions. A star schema design is used to represent Oracle OLAP cubes: dimension views are arranged in a constellation around the cube (or fact) view. Any reporting and analysis tool or application, including complex business intelligence solutions, SQL-based development tools, and Microsoft Excel, can easily and profitably utilize the power of Oracle OLAP thanks to this standard representation of OLAP data.

- SAP BusinessObjects:
  SAP BusinessObjects Business Intelligence serves as a consolidated suite designed for data reporting, visualization, and collaborative sharing. Positioned as the on-premise BI layer within SAP's Business Technology Platform, it converts data into valuable insights accessible at any time and from anywhere. This is like a BI tool with end-to-end functionalities. It is because its features include reporting, analysis, data integration, ETL, Queries, analysis tools, dashboards, Visualization, Enterprise Information Management, Security, adapting to cloud based BI solutions and also the integration with SAP and other systems. Overall it is designed to turn raw data into fruitful insights.

- Microsoft Power BI
  Microsoft Power BI is also a very powerful BI tool with multiple use cases. It is a robust BI tool which incorporates OLAP functionalities which helps us in enhancing data analysis and reporting capabilities. The key features of Power BI is that it allows users to connect to OLAP cubes like Microsoft SQL Server Analysis Services cubes and other multidimensional data sources. I has many functionalities like drag, drop, drill down (ex- matrix table and decomposition tree), filtering, slicing and many more. Users can also create dynamic measures and calculations using DAX in Power BI. This helps us in the creation of KPI’s. Along with this there are many more features in Power BI which we can use and I have personally used them in both DATA-225 and DATA-230 subjects semester end project. I used Power BI to visualize my analysis in the form of a dashboard. According to my personal experience Power BI is easy to use and handle due to the microsoft ecosystem where if we are using Microsoft Teams then we can have sharepoint, Excel, Power BI, Microsoft Azure, Open AI all under one platform.